Connect

$url = **"http://yakyo.innoria.com"**; *//port odoo 8069, bukan postgresql 5432*$db = **"yakyo"**;  
$username = **"admin"**;  
$password = **"1234"**;  
**require\_once**(**'ripcord-master/ripcord.php'**);  
*//hiển thị thông tin version*$common = ripcord::*client*(**"**$url**/xmlrpc/2/common"**);  
$b = $common->version();

Get User ID

$uid = $common->authenticate($db, $username, $password, **array**());

The second endpoint is xmlrpc/2/object, is used to call methods of odoo models via the execute\_kw RPC function.

Bắt buộc phải khởi tạo nếu muốn sử dụng các method của execute\_kw

$models = ripcord::*client*(**"**$url**/xmlrpc/2/object"**);

Search (List ID records)

//Trả về một danh sách id tất cả cac sản phẩm

$models = ripcord::*client*(**"**$url**/xmlrpc/2/object"**);

$ids = $models->execute\_kw($db, $uid, $password,  
 **'product.template'**, **'search'**,  
 **array**(**array**()));

NOTE: product.template là tên của module đang xử lý, nơi chứa dữ liệu

Read Records

// trả về 100 sản phẩm có giá 1234 ,do limit=>1 và list\_price

$ids = $models->execute\_kw($db, $uid, $password,  
 **'product.template'**, **'search'**,  
 **array**(**array**(**array**(**'list\_price'**,**'='**,**'1234'**))),**array**(**'limit'**=>100));

*print\_r*($ids);   
$records = $models->execute\_kw($db, $uid, $password,  
 **'product.template'**, **'read'**, **array**($ids));

Search and Read

//Trả về một danh sách 5 sản phẩm có list\_price với thông tin gồm: name, list\_price, standard\_price do litmit => 5 và array('name', 'list\_price', 'standard\_price').

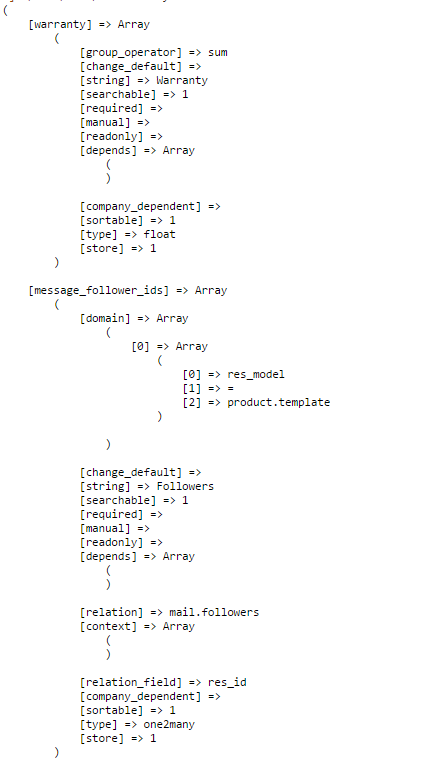
$val\_search\_read = $models->execute\_kw($db, $uid, $password,  
 **'product.template'**, **'search\_read'**,  
 **array**(**array**(**array**(**'list\_price'**,**'='**,**'1'**))),  
 **array**(**'fields'**=>**array**(**'name'**, **'list\_price'**, **'standard\_price'**), **'limit'**=>5));  
*print\_r*($val\_search\_read);

**Fields record**

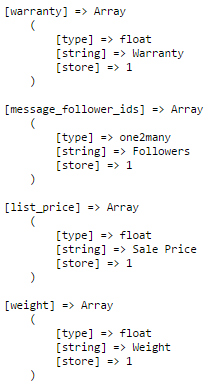
// Trả về dữ liệu là thuộc tính được lọc theo thuộc tính của đối tường ví dụ ở đay là product dữ liệu trả về là các thuộc tính string, type, store của các trường.

$val\_field\_record = $models->execute\_kw($db, $uid, $password,  
 **'product.template'**, **'fields\_get'**, **array**(),**array**(**'attributes'** => **array**(**'string'**,**'type'**,**'store'**)));  
  
*print\_r*($val\_field\_record);

Dữ liệu ban đầu: Ở đây mỗi sản phẩm có nhiều trường thuộc tính như warraty,message\_follower\_ids v..v… Mỗi thuộc tính như warranty hay message\_follower\_ids thì đều có các thuộc tính nhỏ hơn giống nhau như group\_operator, change\_default, string, type , store v.v…



Dữ liệu sau khi lọc sẽ giữ y các trường thuộc tính như warranty, message\_follower\_ids v.v… nhưng những thuộc tính bên trong đã được lược bỏ chỉ còn type, string, store giống như code phía trên



Create Records

// Method tạo thêm record mới bằng cách truyền dữ liệu cần thiết là giá trị của các trường thuộc tính của record qua array. Trong ví dụ này là name, list\_price, standard\_price, image\_medium và image\_small.

$id = $models->execute\_kw($db, $uid, $password,  
 **'product.template'**, **'create'**,  
**array**(**array**(**'name'**=>$name,**'list\_price'**=>$price,**'standard\_price'**=>$cost,**'image\_medium'**=>$imgcode,**'image\_small'**=>$imgcode)));

*print\_r*($id);

Note: Ảnh được truyền cho records là dữ liệu đã được mã hóa thành base64 dạng text .

Với NativeScript : sử dụng module http với phương thức post .. cú pháp http.request

http.request({

url: "http://yakyo.innoria.com/xmlrpc/2/object",

method: "POST",

headers: { "Content-Type": "application/xml" },

content:

"<methodCall><methodName>execute\_kw</methodName><params><param><value><string>yakyo</string></value></param><param><value><int>1</int></value></param><param><value><string>1234</string></value></param><param><value><string>product.template</string></value></param><param><value><string>read</string></value></param><param><value><array><data><value><array><data><value><int>29</int></value></data></array></value></data></array></value></param></params></methodCall>"

// Param read thong tin sp <methodCall><methodName>execute\_kw</methodName><params><param><value><string>yakyo</string></value></param><param><value><int>1</int></value></param><param><value><string>1234</string></value></param><param><value><string>product.template</string></value></param><param><value><string>read</string></value></param><param><value><array><data><value><array><data><value><int>28</int></value><value><int>1</int></value><value><int>12</int></value><value><int>29</int></value></data></array></value></data></array></value></param></params></methodCall>

}).then(function (response) {

var result = response.content.toString(); // response la request tu server . response.content la noi dung can lay, la 1 doi tuong xml

console.log(result);

}, function (e) {

console.log("Error occurred " + e);

});